

Uma Nova Abordagem para Construção de Regras em Redes *Neo-Fuzzy-Neuron* Utilizando Programação Genética Multi-Gene

Glender Brás, Alisson Marques Silva

Programa de Pós-Graduação em Modelagem Matemática e Computacional

Centro Federal de Educação Tecnológica de Minas Gerais - CEFET-MG

Belo Horizonte, MG, Brasil

Email: glenderbras@gmail.com, alisson@cefetmg.br

Resumo—Este artigo propõe a utilização de Programação Genética Multi-Gene (PG-MG) para criação de regras em Redes *Neo-Fuzzy-Neuron* (NFN). NFN é uma rede *Neuro-Fuzzy* composta por n estruturas de Takagi-Sugeno de ordem zero, onde n é o número de variáveis de entrada. A saída individual de cada uma das n estruturas é dada por um conjunto de m regras, em que cada regra é representada por uma função de pertinência Gaussiana. Desta forma, a construção da base de regras na NFN consiste em criar e ajustar as funções de pertinência associadas a cada variável de entrada. A abordagem proposta utiliza a Programação Genética Multi-Gene (PG-MG) para gerar as regras (conjuntos *fuzzy* do antecedente), na qual cada indivíduo da PG-MG representa um conjunto de regras que compõe o modelo. Por fim, os pesos (parâmetros do consequente) são ajustados por um método do Gradiente Descendente. A abordagem proposta é avaliada em problemas de previsão e os resultados obtidos sugerem que a abordagem proposta é competitiva quando comparada com modelos alternativos do estado da arte.

Keywords—*Neo-Fuzzy-Neuron*; Programação Genética Multi-Gene; *Neuro-Fuzzy*.

I. INTRODUÇÃO

A integração entre Sistemas *Fuzzy* e Redes Neurais tem sido amplamente explorada para resolução de diversos problemas, principalmente, no processamento de dados estocásticos e não-lineares [1]. Esta integração tem o objetivo de empregar o tratamento de incerteza e a interpretabilidade dos Sistemas *Fuzzy*, aprimorando sua habilidade de aprendizado através das características presentes nas Redes Neurais [2]. Existe um crescente interesse no desenvolvimento de novos algoritmos de aprendizado que possibilitem a construção autônoma do conjunto de regras de Sistemas *Fuzzy* e *Neuro-Fuzzy* [3]. Dentre as diversas técnicas utilizadas para a construção da base de regras em redes *Neuro-Fuzzy* destaca-se a Computação Evolucionária pela sua capacidade de adaptação e aprendizado baseado em busca global [4].

No trabalho de [5] é utilizado um algoritmo evolucionário multiobjetivo para extração de regras *fuzzy* que descrevam subgrupos do vírus da Influenza A. O algoritmo evolucionário utilizado é baseado no NSGA-II (*Non-dominated Sorting Genetic Algorithm II*), que retorna um pareto ótimo, ou seja, um conjunto de soluções ótimas não-dominadas. Cada

indivíduo representa uma regra e são avaliados por um valor de confiança, de forma que apenas regras com alta confiança sejam consideradas no pareto. Em [6] é proposto um Sistema *Fuzzy* Evolucionário para reconhecimento de atividades em ambientes de saúde e para medir as incertezas associadas a essas atividades. O Algoritmo Genético é utilizado para obter um conjunto de regras ótimas, onde cada indivíduo representa uma regra. Em [7] apresenta-se uma rede *Neuro-Fuzzy* com aprendizado por *Backpropagation* para diagnóstico de risco de hipertensão. Nesse caso o Algoritmo Genético foi utilizado para inicializar a rede *Neuro-Fuzzy* com pesos e vieses otimizados.

A pesquisa de [8] apresenta um protótipo de *software* que utiliza uma rede *Neuro-Fuzzy* para identificar os principais fatores responsáveis pela depressão. Nesse trabalho foi utilizado um Algoritmo Genético (AG) para otimizar os pesos da rede e identificar regras redundantes e inexpressivas, obtendo assim uma base de regras parcimoniosa e otimizada. Utilizando Programação Genética (PG), [9] apresenta um Sistema *Fuzzy* para problemas de classificação. No referido trabalho a Programação Genética foi empregada para criação e ajuste da base de regras de um Sistema *Fuzzy* Genético de Pittsburg. Já em [10] um modelo de PG multi-árvores é empregado para melhorar a acurácia de um Sistema *Fuzzy* de Takagi-Sugeno para mapeamento dinâmico de portfólios.

Neste contexto, este trabalho apresenta um algoritmo baseado em Programação Genética Multi-Gene (PG-MG) para criação da base regras em redes *Neuro-Fuzzy* do tipo *Neo-Fuzzy-Neuron* (NFN) [11]. A rede NFN possui a capacidade de tratar a não linearidade entre os dados através da junção de funções lineares, sendo cada uma delas descritas por regras *fuzzy* [12]. A vantagem deste tipo de rede é o rápido aprendizado, que pode ser até 100 vezes maior que o de uma rede neural multicamada convencional e a garantia de convergência ao ótimo global [13]. Na abordagem proposta cada indivíduo da PG-MG representa uma base de regras. As regras utilizam funções de pertinência Gaussianas no antecedente e são criadas e/ou modificadas pela PG-MG. O treinamento da rede, ou seja, o ajuste dos parâmetros do consequente, é realizado por um método baseado no Gradiente Descendente.

As próximas seções estão organizadas da seguinte forma: a Seção II apresenta a estrutura da rede NFN e o seu processo de aprendizado; na Seção III o algoritmo proposto é apresentado; a Seção IV discorre sobre os experimentos computacionais e discute os resultados obtidos. Por fim, a Seção V traz as considerações finais, discutindo suas contribuições e perspectivas futuras.

II. Neo-Fuzzy-Neuron

Esta seção detalha a rede *Neo-Fuzzy-Neuron* (NFN) [11] utilizada para implementação do algoritmo apresentado na Seção III. *Neo-Fuzzy-Neuron* é uma rede *Neuro-Fuzzy* composta por um conjunto de n modelos do tipo Takagi-Sugeno (TS) [14] de ordem *zero*, um para cada variável de entrada. No NFN a saída de cada modelo individual y_{ti} é dada por um conjunto de m_j regras, e cada regra é representada por uma função de pertinência. A estrutura básica da rede NFN é ilustrada na Figura 1, na qual x_{ti} são as variáveis de entrada no instante t , q_{ij} são os pesos (parâmetros do consequente) associados a cada função de pertinência (regra), y_{ti} é a saída individual de cada modelo e \hat{y}_t é a saída da rede.

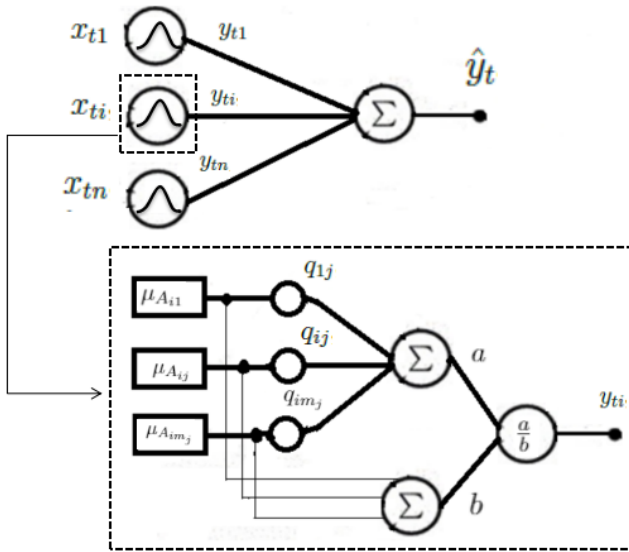


Figura 1. Estrutura da rede Neuro-Fuzzy Neo-Fuzzy-Neuron

A saída do NFN é a soma dos modelos individuais e é obtida no instante t (\hat{y}_t) por:

$$\hat{y}_t = \sum_{i=1}^n y_{ti}. \quad (1)$$

Para cada variável x_{ti} do universo de discurso, a partição do espaço de entrada é realizada por m_j funções de pertinência. A partir dessa partição do espaço de entrada tem-se as seguintes m_j regras:

$$\begin{aligned} R_1 & \text{ Se } x_{ti} \text{ é } A_{i1} \text{ Então } y_{ti} \text{ é } q_{i1} \\ R_2 & \text{ Se } x_{ti} \text{ é } A_{i2} \text{ Então } y_{ti} \text{ é } q_{i2} \\ & \vdots \\ & \vdots \\ R_j & \text{ Se } x_{ti} \text{ é } A_{ij} \text{ Então } y_{ti} \text{ é } q_{ij} \end{aligned}$$

Os modelos são desacoplados e as saídas individuais y_{ti} são obtidas pelo somatório do grau de ativação de cada regra $\mu_{A_{ij}}(x_{ti})$ multiplicado pelo seu respectivo peso q_{ij} (a), ponderados pelo somatório do grau de ativação de todas as regras (b), conforme apresentado na Equação (2):

$$y_{ti} = \frac{\sum_{j=1}^{m_i} \mu_{A_{ij}}(x_{ti}) q_{ij}}{\sum_{j=1}^{m_i} \mu_{A_{ij}}(x_{ti})} = \frac{a}{b} \quad (2)$$

em que, i indexa as variáveis de entrada, j as funções de pertinência e t é o instante de tempo.

As funções de pertinência utilizadas na NFN são Gaussianas. Outros tipos de funções de pertinência também podem ser utilizados, como Triangulares ou Trapezoidais [15], [16], [17]. A escolha do uso de funções de pertinência Gaussiana se justifica por ser o tipo mais adequado para representar dados incertos [18]. Uma função de pertinência Gaussiana (*fpg*) é definida pelo centro (c) e pelo espalhamento (s), como pode ser visto na Figura 2. A Figura 2 ilustra funções de pertinência uniformemente espaçadas e a Figura 3 mostra funções de pertinência não-uniformemente espaçadas.

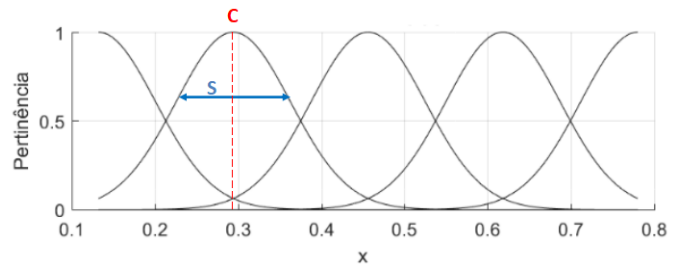


Figura 2. Funções de pertinência Gaussiana uniformemente espaçadas

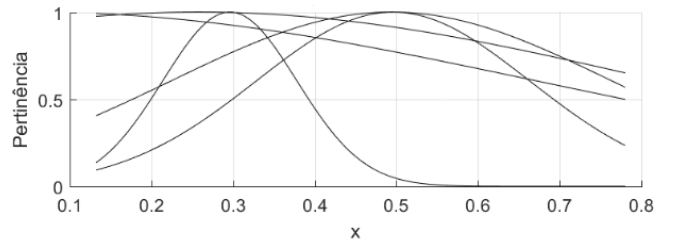


Figura 3. Funções de pertinência Gaussiana

O método de treinamento empregado no algoritmo do NFN é supervisionado e baseado no Gradiente Descendente, no

qual são ajustados os parâmetros q_{ij} . Outros procedimentos, tais como, atualização em lote e atualização incremental com termo de *Momentum*, foram apresentados em [11]. O procedimento de atualização dos pesos descrito a seguir consiste em atualizar os pesos a cada época de treinamento a partir de uma taxa de aprendizado (α) previamente definida. A fórmula para atualização dos pesos pode ser vista na Equação (3):

$$q_{ij} = q_{ij} - \alpha(e_t)(x_{ti})(d_{ij}), \quad (3)$$

na qual α é a taxa de aprendizado, e_t é o erro de modelagem obtido por $y_t - \hat{y}_t$, x_{ti} é a amostra de dados no instante t para a variável i e d_{ij} é obtido por:

$$d_{ij} = \frac{\mu_{A_{ij}}}{\sum_{j=1}^{m_i} \mu_{A_{ij}}}, \quad (4)$$

onde m_i é o número de funções de pertinência para a variável i .

O Algoritmo 1 sumariza a estimação da saída e o processo de atualização dos parâmetros da rede NFN.

```

Entrada  $x_t, y_t, n$ ;
Saída  $\hat{y}_t$ ;
Inicializar  $fp_g$ ;
for  $epoca = 1 : l$  do
    for  $t = 1, 2, \dots$  do
        Ler  $x_t, y_t$ ;
        for  $j = 1 : m$  do
            for  $i = 1 : n$  do
                Calcular  $\mu_{A_{ij}}$ ;
                Calcular  $y_{ti}$ ;
                Calcular  $d_{ij}$ ;
                Atualizar  $q_{ij}$ ;
            end
        end
    end
    Calcular  $\hat{y}_t$ ;
    Calcular  $e_t$ ;
end

```

Algorithm 1: Algoritmo da Rede *Neo-Fuzzy-Neuron*

III. NFN-PG-MG - *Neo-Fuzzy-Neuron* com Programação Genética Multi-Gene

Esta seção introduz a abordagem proposta para construção da base de regras da rede *Neo-Fuzzy-Neuron*. A metodologia utilizada considera que um indivíduo representa uma base de regras completa [19] para a rede NFN e aplica Programação Genética Multi-Gene (PG-MG) [20] para modelagem dos indivíduos. Por fim, um método baseado no Gradiente Descendente é empregado para ajuste dos pesos da rede.

A PG-MG é uma variação da Programação Genética que considera que um indivíduo possui em sua estrutura várias árvores conectadas por uma estrutura linear de genes superiores. Cada gene superior é uma estrutura que sintetiza o

resultado de uma das árvores que compõe o indivíduo e a saída final é a agregação do resultado de cada gene superior. No modelo apresentado neste trabalho, cada gene superior representa um modelo individual do NFN. Cada um desses modelos está associado a uma árvore de aridade m , sendo m o número de regras, ou seja, cada sub-árvore ligada ao nó central representa uma regra (função de pertinência). O nó raiz é composto por uma função de agregação que retorna o resultado das m regras associadas àquele modelo individual, conforme discutido na Seção II e ilustrado na Figura 1.

O Cruzamento na PG-MG pode ser realizado em alto nível ou baixo nível. O Cruzamento em alto nível ocorre quando troca-se um gene superior de um pai com um gene superior do outro. Quando o Cruzamento ocorre nas sub-árvores associadas ao gene superior, chama-se Cruzamento de baixo nível. Neste trabalho foi utilizado Cruzamento de baixo nível, combinando-se apenas as sub-árvores que representam as regras. Informações complementares sobre Cruzamento de alto e baixo nível podem ser encontradas em [9]. A Reprodução é realizada gerando um filho exatamente igual ao pai (clone). Já a Mutação altera apenas uma função (sub-árvore) de um gene sorteado, trocando-a por uma nova função. O método de seleção de pais é o método do torneio [21] e a geração da nova população é realizada substituindo-se toda a população atual pela nova população gerada.

A abordagem proposta para criação/ajuste da base de regras e treinamento do NFN-PG-MG pode ser visualizado graficamente na Figura 4 e é sumarizado nos seguintes passos:

- 1) A primeira etapa do algoritmo é particionar o universo de domínio de cada variável de entrada em fp_g funções de pertinência Gaussianas uniformemente espaçadas. Em outras palavras, cria-se o primeiro indivíduo da população. A variável fp_g é um parâmetro de entrada do algoritmo.
- 2) Os pesos q_{ij} são iniciados com valores randômicos entre 0 e 1.
- 3) Após a inicialização dos pesos da rede são executadas p épocas de treinamento para ajuste dos pesos utilizando o algoritmo baseado no Gradiente Descendente descrito na Seção II. Experimentos computacionais realizados mostram que os melhores resultados foram obtidos com o número de épocas de treinamento com valores de p entre 5 e 10. Note que o valor de p pode ser definido como 0 caso opte por não realizar a etapa de ajuste de pesos inicial e utilizar as funções apenas para inicializar a PG-MG para a próxima etapa. Destaca-se que o objetivo desta etapa é melhorar a inicialização dos pesos e não obter pesos ótimos.
- 4) A quarta etapa é a geração da população inicial. A população inicial da PG-MG é criada com o primeiro indivíduo descrito no Passo 1 e com os demais indivíduos gerados com o número de funções de pertinência geradas entre o número mínimo (min_{FP}) e o máximo (max_{FP}) de funções de pertinência. Sendo min_{FP} e max_{FP} parâmetros de entrada do algoritmo. As funções Gaussianas são definidas sorteando-se, para cada função,

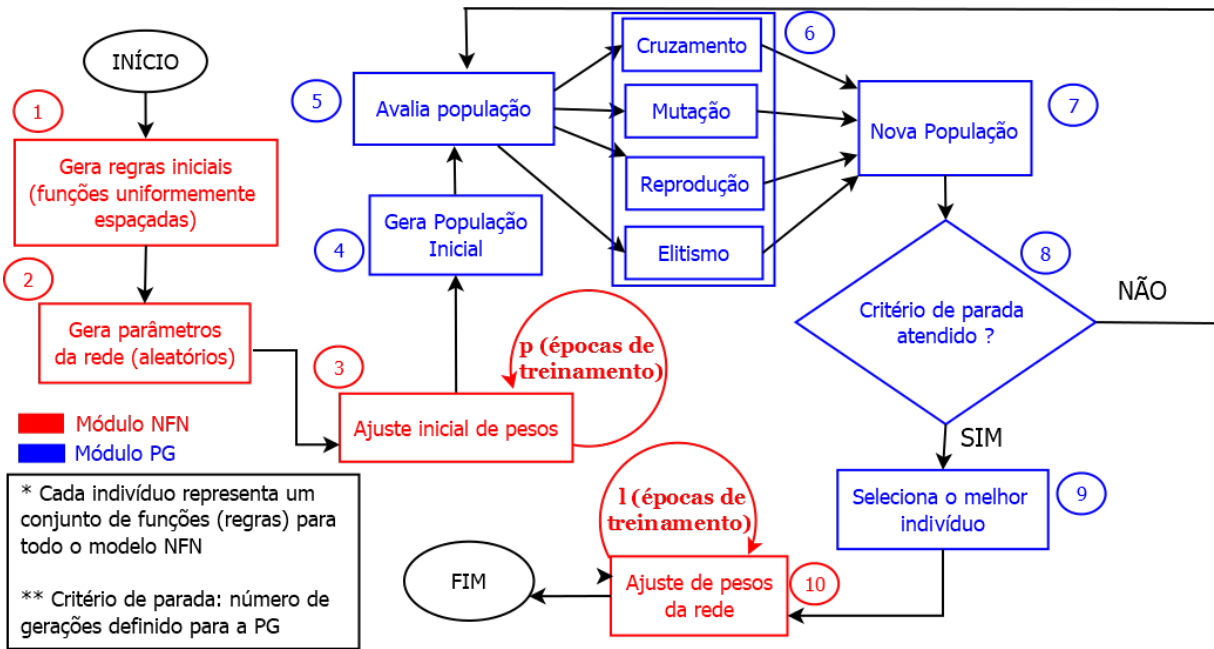


Figura 4. Fluxograma do algoritmo NFN-PG-MG

um valor de c (centro da função) entre o limite inferior e o limite superior do domínio da variável de entrada, e um valor para o parâmetro s (espalhamento).

- 5) Na quinta etapa avalia-se a população gerada calculando o erro de modelagem de cada indivíduo (conjunto de regras). Os indivíduos são avaliados pelo Erro Quadrático Médio (EQM) obtido por:

$$EQM = \frac{\sum_{t=1}^k (e_t^2)}{k}, \quad (5)$$

onde e_t é o erro (diferença entre a saída esperada y_t e a saída do modelo \hat{y}_t) no instante t e k é o número de amostras.

- 6) Considerando o resultado da avaliação realizada na Etapa 5 o melhor indivíduo passará pela próxima geração por Elitismo e, os demais indivíduos da nova população são obtidos pelas operações de Cruzamento, Mutaçao e Reproduçao. Os pais são selecionados pelo método do torneio [21], onde é realizado um torneio entre uma quantidade determinada de indivíduos da população para selecionar cada pai, sendo escolhido o indivíduo com melhor performance entre estes. Na abordagem utilizada, os filhos gerados sempre substituem os pais, podendo existir indivíduos da nova população iguais aos da população anterior devido ao operador de Reproduçao, que gera filhos exatamente iguais aos pais. No Cruzamento é sorteado um ponto de corte para cada árvore ligada a cada gene superior de cada um dos pais, e os filhos são gerados recombinando as informações a partir deste ponto. Na abordagem proposta, cada gene superior representa um modelo individual do NFN, ou

seja, é realizado o Cruzamento em n pontos, sendo n o número de variáveis de entrada da rede, permitindo assim a alteração de regras referentes a todas as variáveis em cada Cruzamento realizado.

O operador de Cruzamento é o responsável por modificar o número de funções de pertinência dos indivíduos pai para os indivíduos filho, visto que os dois pais selecionados podem ter quantidade de funções diferentes e o ponto de Cruzamento sorteado também pode não ser o mesmo para os dois, desta forma, os pais poderão gerar filhos com número de funções distintas. Já na operação de Mutaçao, é escolhida apenas uma função referente a um gene superior para ser alterada. Ou seja, é realizada Mutaçao de 1 ponto, sendo alterada apenas uma regra referente a uma variável de entrada.

Embora não seja comum, caso a melhor configuração para a rede NFN seja com funções uniformemente espaçadas, todas as outras configurações geradas pela PG-MG serão inferiores quanto à performance e as funções iniciais permanecerão até a última geração. Caso não seja a melhor configuração, esse indivíduo com funções uniformemente espaçadas poderá influenciar na geração das próximas bases de regras, conforme sua medida de desempenho.

- 7) Nesta etapa a população atual é substituída pela nova população gerada na Etapa 6.
 8) Repete-se as Etapas 5, 6 e 7 até atingir o número de gerações estabelecido como critério de parada. Opcionalmente, pode-se utilizar como critério de parada uma medida de erro como EQM (Equação 5) ou RMSE (Equação 8), por exemplo.

- 9) Após a execução de todas as gerações da PG-MG o melhor indivíduo (indivíduo com menor erro) é selecionado e terá seus pesos ajustados na etapa final.
- 10) Por fim, a última etapa consiste no ajuste de pesos do melhor indivíduo (conjunto de regras/funções de pertinência) definidos pela Programação Genética Multi-Gene. Esse ajuste é realizado por um método baseado no Gradiente Descendente com l épocas de treinamento.

Note que, após a etapa de treinamento (construção da base de regras e ajuste dos pesos) todos os parâmetros são fixados e, então, utilizados para realizar a previsão/identificação de sistema.

IV. EXPERIMENTOS COMPUTACIONAIS

O NFN-PG-MG foi avaliado e comparado com modelos alternativos em problemas de previsão. Os modelos alternativos utilizados nos experimentos foram o ANFIS (*Adaptive Neuro-Fuzzy Inference System*) com aprendizado por Mínimos Quadrados e Gradiente Descendente com *Backpropagation*, uma Rede Neural MLP (*Multilayer Perceptron*) com aprendizado pelo método do Gradiente Descendente com *Backpropagation* e uma rede *Neo-Fuzzy-Neuron* tradicional com aprendizado pelo método do Gradiente.

Para cada experimento o conjunto de dados foi dividido em três subconjuntos, um para treinamento com 60% das amostras, um para validação com 20% e outro para teste com 20% das amostras. Os experimentos foram realizados *off-line*, isto é, após a etapa de treinamento todos os parâmetros dos são fixados e utilizados para realizar o experimento no subconjunto de teste. Cada experimento foi executado 10 vezes e o melhor resultado foi utilizado para comparação. O desempenho dos sistemas foi avaliado pelo *RMSE (Root Mean Square Error)* (8):

$$RMSE = \frac{1}{k} \left(\sum_{t=1}^k (y_t - \hat{y}_t) \right)^{\frac{1}{2}}, \quad (6)$$

onde k é o número de amostra do conjunto de teste, \hat{y}_t é a saída estimada, y_t é a saída desejada. Os parâmetros dos modelos foram definidos como descrito a seguir para os dois experimentos realizados:

- ANFIS: épocas de treinamento = 500; funções de pertinência inicial = 2; tipo das funções pertinência = gaussiana; $\alpha = 0.01$.
- NFN-PG-MG: tamanho da população = 50; gerações = 300; $p = 5$; épocas de treinamento (l) = 200; taxa de cruzamento = 0.9; taxa de mutação = 0.08; $\alpha = 0.01$; funções de pertinência inicial = 5; máximo de funções de pertinência = 10; tipo das funções de pertinência = gaussiana.
- NFN: épocas de treinamento = 500; $\alpha = 0.01$; funções de pertinência = 5; tipo das funções de pertinência = gaussiana.
- RNA MLP: neurônios por camada = 10; épocas de treinamento = 500; $\alpha = 0.01$; função de ativação tangente

hiperbólica. Foram realizados experimentos com 2 e 3 camadas ocultas.

A. Previsão de Temperatura

Esta seção visa avaliar os modelos para previsão de temperatura em três regiões geográficas com padrões climáticos distintos. Considera-se os dados referentes a temperatura média mensal em Ottawa: região de clima frio; Vale da Morte: região com clima extremamente seco e altas temperaturas; e Lisboa região com clima bastante variado ao longo do ano, oscilando de um inverno bastante frio com neve a um verão quente. O objetivo é prever a temperatura média mensal um passo à frente. Trabalhos anteriores sugerem utilizar os cinco primeiros valores defasados da série como entrada [22], [23]. Desta forma, o modelo pode ser descrito por:

$$y_t = f(y_{t-1}, y_{t-2}, y_{t-3}, y_{t-4}, y_{t-5}). \quad (7)$$

Serão apresentados 3 experimentos, 1 para cada região. Os dados estão normalizados entre 0 e 1.

1) *Ottawa*: Neste experimento consideram-se os dados de temperatura média mensal da região de Ottawa no período de janeiro de 1895 a dezembro de 2009. O conjunto de dados é composto por 1374 amostras, sendo 824 para treinar os modelos, 275 para validar e 275 para avaliar seu desempenho pelo RMSE. Os resultados obtidos pelos modelos para previsão de temperatura em Ottawa são exibidos na Tabela I que mostra que o melhor resultado foi obtido pelo NFN-PG-MG. Os resultados obtidos pelo NFN-PG-MG, NFN e ANFIS são comparáveis e superiores aos obtidos pelas redes MLP com duas e com três camadas. Os valores desejados e os estimados pelo NFN-PG-MG para os dados de avaliação podem ser vistos na Figura 5.

Tabela I
DESEMPENHO DO NFN-PG-MG NA PREVISÃO DE TEMPERATURA EM OTTAWA

| Modelo | RMSE |
|----------------|------------|
| NFN-PG-MG | 0,03384693 |
| NFN | 0,03425178 |
| ANFIS | 0,03768260 |
| MLP- 2 camadas | 0,07574512 |
| MLP- 3 camadas | 0,08157002 |

2) *Vale da Morte*: Neste experimento consideram-se as informações de temperatura média mensal da região do Vale da Morte no período de janeiro de 1901 a dezembro de 2009. O conjunto de amostra possui 1302 amostras, sendo 781 utilizadas para treinar os modelos, 260 para validar e 261 para avaliar seu desempenho pelo RMSE. Os resultados obtidos pelos modelos para previsão de temperatura no Vale da Morte são exibidos na Tabela II. Para esta base de dados, o melhor desempenho foi do NFN tradicional seguido pelo NFN-PG-MG. Os resultados obtidos pelo NFN tradicional e pelo NFN-PG-MG são comparáveis e superam os obtidos pelos demais modelos. Os valores desejados e os estimados pelo NFN-PG-MG para os dados de avaliação podem ser vistos na Figura 6.

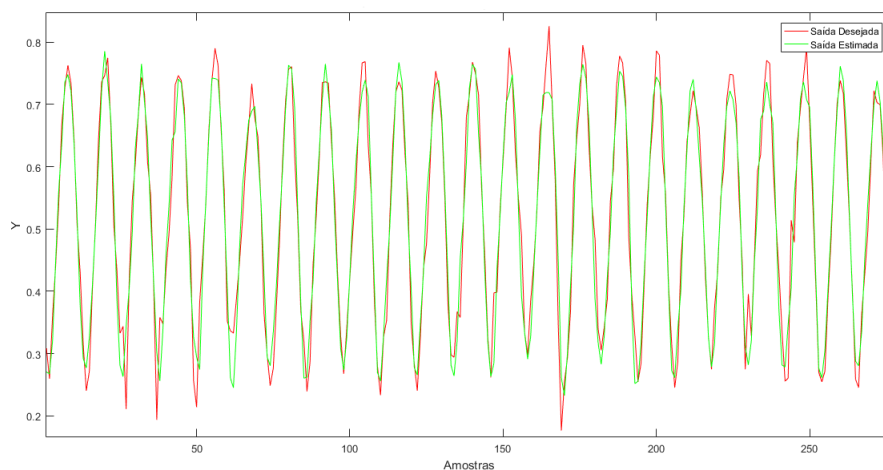


Figura 5. Previsão de temperatura em Ottawa pelo NFN-PG-MG

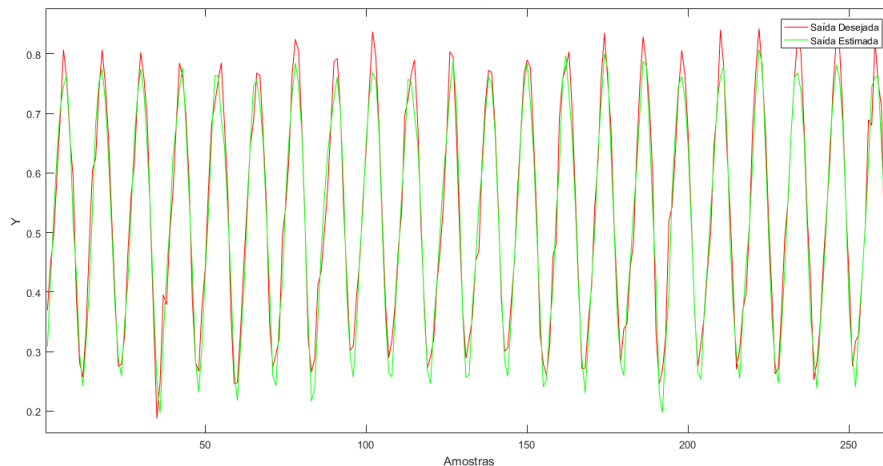


Figura 6. Previsão de temperatura no Vale da Morte pelo NFN-PG-MG

Tabela II
DESEMPENHO DO NFN-PG-MG NA PREVISÃO DE TEMPERATURA NO VALE DA MORTE

| Modelo | RMSE |
|-----------------|------------|
| NFN | 0,02440318 |
| NFN-PG-MG | 0,02502818 |
| ANFIS | 0,03008540 |
| MLP - 2 camadas | 0,06066379 |
| MLP - 3 camadas | 0,06290018 |

3) *Lisboa*: Para a região de Lisboa o conjunto de dados se refere à temperatura média mensal no período de janeiro de 1910 a dezembro de 2009 e é composto por 1194 amostras, sendo 716 para treinar os modelos, 239 para validação e 239 para avaliar seu desempenho pelo RMSE. Os resultados obtidos pelos modelos para previsão de temperatura em Lisboa

são exibidos na Tabela III. Os resultados observados no NFN tradicional e no NFN-PG-MG se demonstraram competitivos e superam os obtidos pelos outros modelos alternativos. Os valores desejados e os estimados pelo NFN-PG-MG para os dados de avaliação podem ser vistos na Figura 7.

Tabela III
DESEMPENHO DO NFN-PG-MG NA PREVISÃO DE TEMPERATURA EM LISBOA

| Modelo | RMSE |
|-----------------|------------|
| NFN | 0,03460898 |
| NFN-PG-MG | 0,03518216 |
| ANFIS | 0,04266740 |
| MLP - 2 camadas | 0,07752571 |
| MLP - 3 camadas | 0,08176709 |

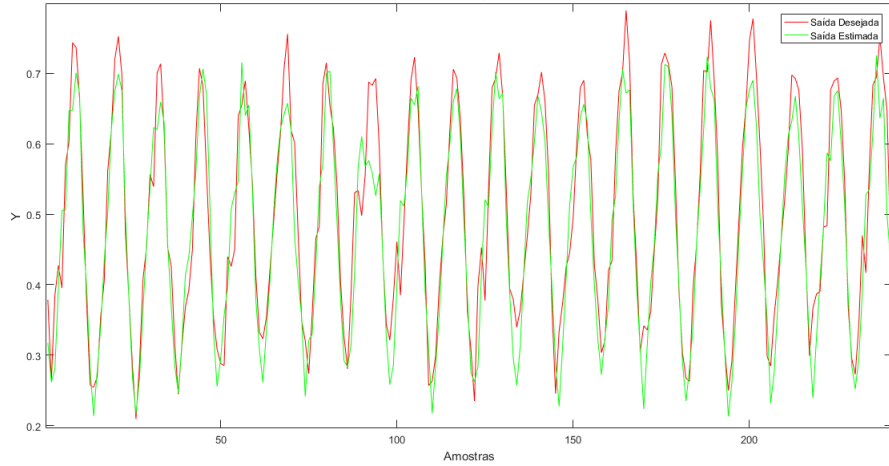


Figura 7. Previsão de temperatura em Lisboa pelo NFN-PG-MG

B. Série Temporal de Mackey-Glass

Nesta seção os modelos são avaliados para previsão da série temporal clássica de Mackey-Glass [24]. A série é criada pela seguinte equação diferencial com atraso:

$$\frac{dx(t)}{dt} = \frac{0.2x(t-\tau)}{1+x(t-\tau)^{10}} - 0.1x(t), \quad (8)$$

onde $x(0) = 1.2$ e $\tau = 17$. O objetivo é prever o valor x_{t+6} para qualquer valor de t baseado nas entradas $[x_{t-18}x_{t-12}x_{t-6}x_t]$. O conjunto de dados é composto de 3500 amostras, com 4 variáveis de entrada e 1 de saída, das quais 3000 forma coletadas com $t \in [201, 3200]$ e 500 com $t \in [5001, 5500]$ [25]. Foram utilizadas 2100 para treinar os modelos, 700 para validar e 700 para avaliar o desempenho dos modelos pelo RMSE.

A Tabela IV mostra o desempenhos dos modelos para previsão da série temporal de Mackey-Glass. Observa-se que o melhor desempenho foi obtido pelo modelo ANFIS seguido pelo NFN-PG-MG. O NFN-PG-MG demonstrou desempenho melhor inclusive que o NFN tradicional. A Figura 8 ilustra os valores desejados e os estimados pelo NFN-PG-MG para os dados de avaliação.

Tabela IV
DESEMPENHO DO NFN-PG-MG NA PREVISÃO DA SÉRIE TEMPORAL DE
MACKEY-GLASS

| Modelo | RMSE |
|---------------|------------|
| ANFIS | 0,01851480 |
| NFN-PG-MG | 0,05548425 |
| NFN | 0,06019720 |
| MLP-3 camadas | 0,11302678 |
| MLP-2 camadas | 0,19026832 |

V. CONSIDERAÇÕES FINAIS

Este trabalho introduziu um novo algoritmo baseado em Programação Genética Multi-Gene para construção da base de redes em redes *Neo-Fuzzy-Neuron*. A abordagem utiliza a Programação Genética Multi-Gene para criação e ajuste da base de regras e um método baseado no Gradiente para a ajuste de pesos.

Experimentos computacionais em problemas de previsão sugerem que a abordagem seja promissora. O NFN-PG-MG se demonstrou eficiente quando comparado a modelos alternativos, sendo sempre superior a pelo menos dois dos três modelos utilizados para comparação. Além disso, observa-se que em relação ao NFN tradicional os resultados, quando não superiores, são bastante próximos e competitivos. Desta forma, a Programação Genética se demonstrou uma boa técnica para modelagem de redes NFN para problemas de previsão e identificação de sistemas. Além da acurácia, destaca-se na abordagem proposta a construção autônoma e otimizada da base de regras de acordo com o conjunto de dados.

Trabalhos futuros devem explorar a utilização da Programação Genética também para ajuste dos pesos, modelando as funções com seus pesos como indivíduos da Programação Genética. Além disso pode-se utilizar métodos de Programação Genética Semântica, que visa mitigar alguns problemas que podem ser encontrados pela Programação Genética tradicional. Também pode-se considerar utilizar funções de pertinência triangulares na rede NFN.

AGRADECIMENTOS

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001.

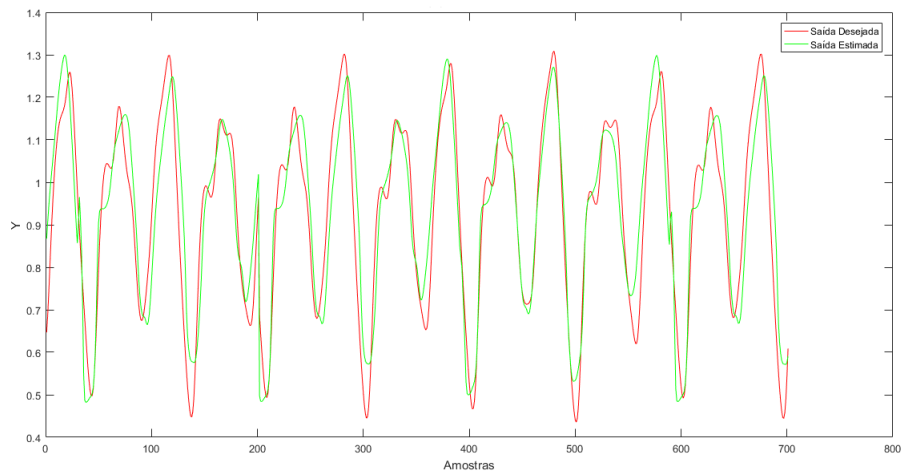


Figura 8. Previsão da série temporal de Mackey-Glass pelo NFN-PG-MG

REFERÊNCIAS

- [1] Y. Bodyanskiy, I. Pliss, and O. Vynokurova, "Flexible neo-fuzzy neuron and neuro-fuzzy network for monitoring time series properties," *Information Technology and Management Science*, vol. 16, no. 1, pp. 47–52, 2013.
- [2] W. Pedrycz, "Neurocomputations in relational systems," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 13, no. 3, pp. 289–297, Mar. 1991. [Online]. Available: <http://dx.doi.org/10.1109/34.75517>
- [3] K. Shihabudheen and G. Pillai, "Recent advances in neuro-fuzzy system," *Know-Based Syst.*, vol. 152, no. C, pp. 136–162, Jul. 2018. [Online]. Available: <https://doi.org/10.1016/j.knosys.2018.04.014>
- [4] F. Herrera, "Genetic fuzzy systems: taxonomy, current research trends and prospects," *Evolutionary Intelligence*, vol. 1, no. 1, pp. 27–46, 2008.
- [5] C. J. Carmona, C. Chrysostomou, H. Seker, and M. del Jesus, "Fuzzy rules for describing subgroups from influenza a virus using a multi-objective evolutionary algorithm," *Applied Soft Computing*, vol. 13, no. 8, pp. 3439–3448, 2013.
- [6] M. Fahim, I. Fatima, S. Lee, and Y.-T. Park, "Efm: evolutionary fuzzy model for dynamic activities recognition using a smartphone accelerometer," *Applied intelligence*, vol. 39, no. 3, pp. 475–488, 2013.
- [7] A. Kaur, A. Bhardwaj, and U. A. H. Been, "Genetic neuro fuzzy system for hypertension diagnosis," *Heart*, vol. 19, p. 25, 2014.
- [8] K. Ashish, A. Dasari, S. Chattopadhyay, and N. B. Hui, "Genetic-neuro-fuzzy system for grading depression," *Applied computing and informatics*, vol. 14, no. 1, pp. 98–105, 2018.
- [9] A. S. Koshiyama, M. M. Vellasco, and R. Tanscheit, "Gpfs-class: A genetic fuzzy system based on genetic programming for classification problems," *Appl. Soft Comput.*, vol. 37, no. C, pp. 561–571, Dec. 2015. [Online]. Available: <https://doi.org/10.1016/j.asoc.2015.08.055>
- [10] S. Mousavi, A. Esfahanipour, and M. H. F. Zarandi, "Mgp-intactsky: Multitree genetic programming-based learning of interpretable and accurate tsk systems for dynamic portfolio trading," *Applied Soft Computing*, vol. 34, pp. 449 – 462, 2015. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S156849461500321X>
- [11] T. Yamakawa, E. Uchino, T. Miki, and H. Kusabagi, "A neo fuzzy neuron and its applications to system identification and predictions to system behavior," in *Proceedings of the International Conference on Fuzzy Logic and Neural Networks*. IEEE, 1992, pp. 477–484.
- [12] T. Miki and T. Yamakawa, "Analog implementation of neo-fuzzy neuron and its on-board learning," *Computational Intelligence and Applications*, pp. 144–149, 1999.
- [13] T. Yamakawa, H. Kusanagi, E. Uchino, and T. Miki, "A new effective learning algorithm for a neo fuzzy neuron model," in *Proceedings of the Korean Institute of Intelligent Systems Conference*. Korean Institute of Intelligent Systems, 1993, pp. 1017–1020.
- [14] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-15, no. 1, pp. 116–132, Jan 1985.
- [15] Y. Bodyanskiy, I. Kokshenev, and V. Kolodyazhnyi, "An adaptive learning algorithm for a neo fuzzy neuron," in *EUSFLAT Conf.* Citeseer, 2003, pp. 375–379.
- [16] Y. Zaychenko and A. Gasanov, "Investigations of cascade neo-fuzzy neural networks in the problem of forecasting at the stock exchange," in *2012 IV International Conference "Problems of Cybernetics and Informatics" (PCI)*, Sep. 2012, pp. 1–3.
- [17] A. M. Silva, W. Caminhas, A. Lemos, and F. Gomide, "A fast learning algorithm for evolving neo-fuzzy neuron," *Appl. Soft Comput.*, vol. 14, pp. 194–209, Jan. 2014. [Online]. Available: <https://doi.org/10.1016/j.asoc.2013.03.022>
- [18] V. Kreinovich, C. Quintana, and L. Reznik, "Gaussian membership functions are most adequate in representing uncertainty in measurements," in *Proceedings of NAFIPS*, 1992, pp. 15–17.
- [19] S. F. Smith, "A learning system based on genetic adaptive algorithms," Ph.D. dissertation, Pittsburgh, PA, USA, 1980, aAI8112638.
- [20] C. Ferreira, "Gene expression programming: a new adaptive algorithm for solving problems," *Complex Systems*, vol. 13, no. 2, pp. 87–129, 2001, cite arxiv:cs/0102027Comment: 22 pages, 17 figures. [Online]. Available: <http://arxiv.org/abs/cs/0102027>
- [21] R. Poli, W. B. Langdon, and N. F. McPhee, *A Field Guide to Genetic Programming*. Lulu Enterprises, UK Ltd, 2008.
- [22] D. Leite, R. Ballini, P. Costa, and F. Gomide, "Evolving fuzzy granular modeling from nonstationary fuzzy data streams," *Evolving Systems*, vol. 3, no. 2, pp. 65–79, Jun 2012. [Online]. Available: <https://doi.org/10.1007/s12530-012-9050-9>
- [23] A. M. Silva, W. M. Caminhas, A. P. Lemos, and F. Gomide, "Evolving neural fuzzy network with adaptive feature selection," in *2012 11th International Conference on Machine Learning and Applications*, vol. 2, Dec 2012, pp. 440–445.
- [24] M. Mackey and L. Glass, "Oscillation and chaos in physiological control systems," *Science*, vol. 197, no. 4300, pp. 287–289, 1977. [Online]. Available: <https://science.sciencemag.org/content/197/4300/287>
- [25] N. K. Kasabov and Qun Song, "Denfis: dynamic evolving neural-fuzzy inference system and its application for time-series prediction," *IEEE Transactions on Fuzzy Systems*, vol. 10, no. 2, pp. 144–154, April 2002.